# SUPPLEMENTARY FILE

*OpenPrescribing: Normalised Data and Software Tool to Research Trends in English NHS Primary Care Prescribing 1998-2016*

## Contents:

**Table S1.** Summary of original (unprocessed) PCA data by year (drugs in Chapters 1-15 only).

**Table S2.** Summary of drug code changes within the 1998-2016 prescribing datasets, by year.

**Figure S1.** Screenshots from Trends tool, showing items per 1,000 population and inflation-corrected costs per 1,000 population for selected drugs.

**Appendix** - SQL Code for Processing and Normalisation of PCA data.

**Table S1.** Summary of original (unprocessed) PCA data by year (drugs in Chapters 1-15 only).

| Year | Distinct count of Drug Name | Items | Cost |
|---|---|---|---|
| 1998 | 6,338 | 497M | £4,440M |
| 1999 | 6,587 | 513M | £5,011M |
| 2000 | 6,613 | 535M | £5,284M |
| 2001 | 6,754 | 569M | £5,784M |
| 2002 | 6,834 | 599M | £6,487M |
| 2003 | 6,893 | 630M | £7,113M |
| 2004 | 6,912 | 666M | £7,645M |
| 2005 | 6,907 | 699M | £7,452M |
| 2006 | 6,810 | 728M | £7,660M |
| 2007 | 7,056 | 772M | £7,810M |
| 2008 | 7,202 | 817M | £7,716M |
| 2009 | 7,401 | 859M | £7,892M |
| 2010 | 11,703 | 898M | £8,162M |
| 2011 | 11,751 | 932M | £8,101M |
| 2012 | 12,207 | 969M | £7,802M |
| 2013 | 12,318 | 996M | £7,846M |
| 2014 | 12,576 | 1,027M | £8,022M |
| 2015 | 12,875 | 1,043M | £8,403M |
| 2016 | 13,285 | 1,061M | £8,284M |
| **Total** | ***27,473*** | **14,811M** | **£136,914M** |

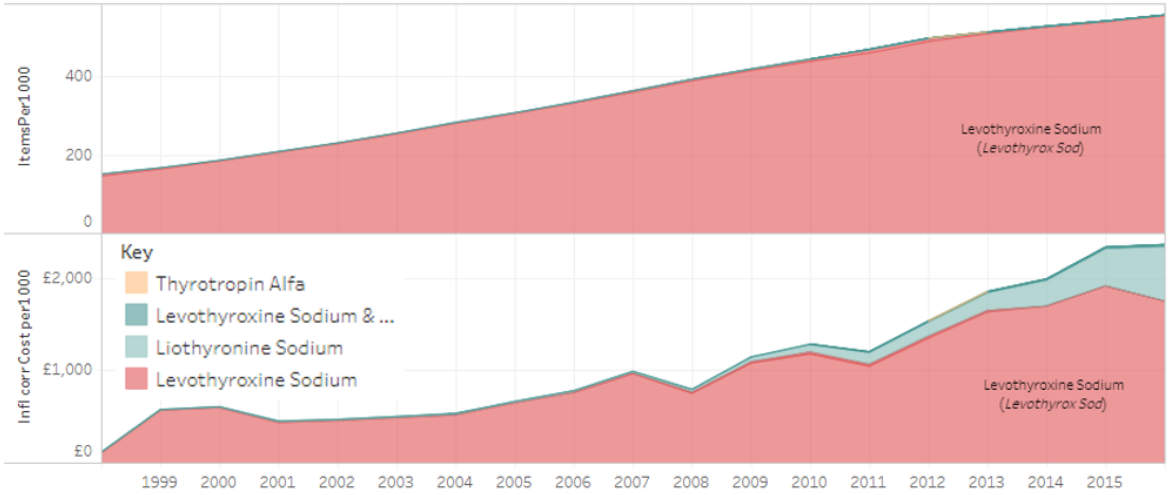
**Table S2.** Summary of drug code changes within the 1998-2016 prescribing datasets, by year.

| | Code change | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| year | Chapter | Section | Paragraph | Sub-Paragraph | no change | no product match | no chemical match | Grand Total |
| 1998 | 21 | 9 | 219 | 140 | 5,609 | 230 | 110 | **6,338** |
| 1999 | 21 | 7 | 188 | 88 | 5,917 | 237 | 129 | **6,587** |
| 2000 | 22 | 8 | 189 | 86 | 5,928 | 260 | 120 | **6,613** |
| 2001 | 26 | 9 | 178 | 85 | 6,069 | 278 | 111 | **6,756** |
| 2002 | 31 | 10 | 182 | 77 | 6,173 | 274 | 87 | **6,834** |
| 2003 | 34 | 9 | 191 | 79 | 6,256 | 241 | 83 | **6,893** |
| 2004 | 36 | 16 | 191 | 77 | 6,265 | 240 | 87 | **6,912** |
| 2005 | 26 | 14 | 157 | 85 | 6,315 | 245 | 65 | **6,907** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2006 | 27 | 17 | 158 | 72 | 6,245 | 232 | 59 | **6,810** |
| 2007 | 35 | 12 | 168 | 74 | 6,483 | 237 | 47 | **7,056** |
| 2008 | 34 | 12 | 169 | 74 | 6,648 | 231 | 34 | **7,202** |
| 2009 | 3 | 4 | 180 | 76 | 6,919 | 219 | | **7,401** |
| 2010 | 12 | 8 | 207 | 14 | 11,182 | 279 | 1 | **11,703** |
| 2011 | 15 | 4 | 130 | 14 | 11,331 | 257 | | **11,751** |
| 2012 | 8 | 4 | 4 | 16 | 11,946 | 229 | | **12,207** |
| 2013 | 2 | 4 | 8 | 1 | 12,139 | 163 | 1 | **12,318** |
| 2014 | | 6 | 15 | 1 | 12,415 | 139 | | **12,576** |
| 2015 | | 6 | 6 | | 12,770 | 93 | | **12,875** |
| 2016 | | 6 | 7 | | 13,241 | 31 | | **13,285** |

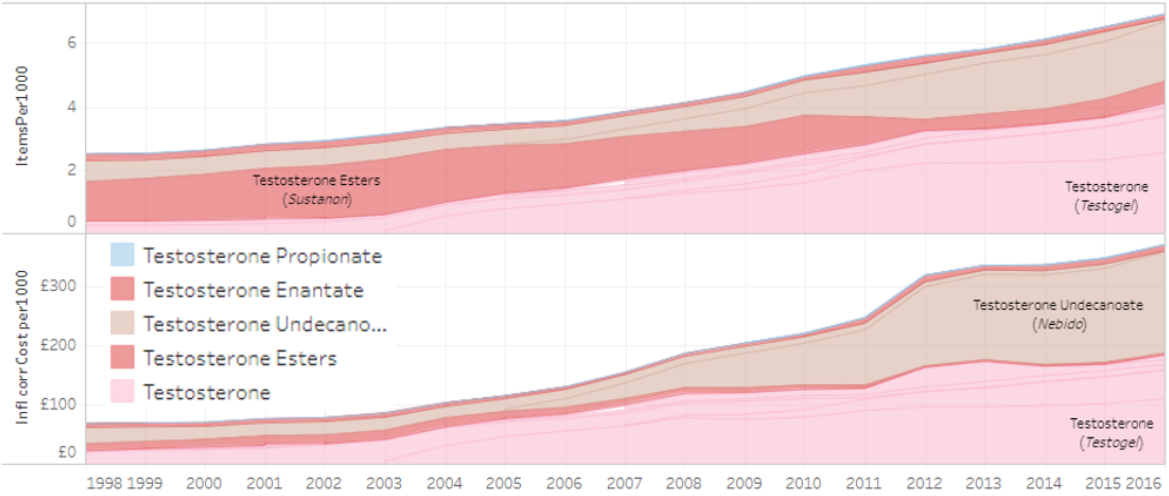
a

Items Dispensed and Costs per 1000 population for all products of Chemical: Thyrotropin Alfa, Levothyroxine Sodium & Liothyronine, Liothyronine Sodium and 1 more, in paragraph: Thyroid Hormones




b

Items Dispensed and Costs per 1000 population for all products of Chemical: Testosterone Propionate, Testosterone Enantate, Testosterone Undecanoate and 2 more, in paragraph: Male Sex Hormones And Antagonists




c

Items Dispensed and Costs per 1000 population for all products of Chemical: Quinine Dihydrochloride, Quinine Hydrochloride, Quinine Bisulfate and 1 more, in paragraph: Antimalarials

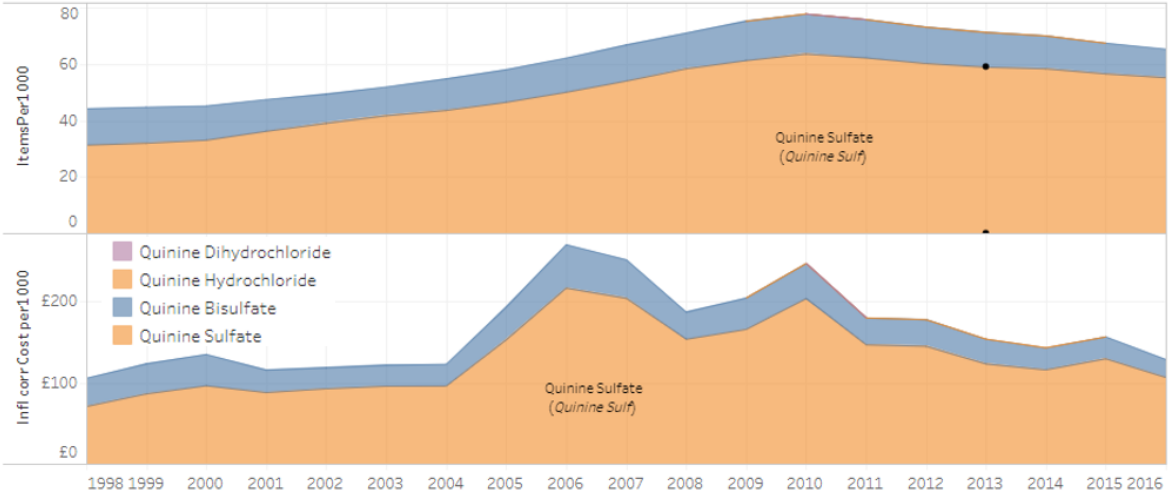


**Figure S1.** Screenshots from Trends tool, showing items per 1,000 population and inflation-corrected costs per 1,000 population for selected drugs. Product names are in parentheses. (a) Prescribing trends for all chemicals within the Paragraph of Thyroid Hormones. Full dashboard available at https://public.tableau.com/shared/GPW28PWJY?:display_count=yes. (b) Prescribing trends for all testosterone chemicals within the Paragraph of Male Sex Hormones. Full dashboard available at https://public.tableau.com/shared/YQ3ZFB3HY?:display_count=yes. c) Prescribing trends for all chemical forms of quinine (all of which are in the Antimalarials Paragraph). Full dashboard available at https://public.tableau.com/shared/85KJ2ZFN4?:display_count=yes.

## Appendix - SQL Code for Processing and Normalisation of PCA data

### *A - Lookup Tables*

A1. The special_cases lookup table

This is a workaround to assign a 'most likely' classification to the few problematic drug names which exist multiple times in BNF.

Lookup table is created by running the following script:

```
WITH temp as (
SELECT SUBSTR(SECTION_CODE,1,2) as chapter, section_code, presentation,
COUNT(DISTINCT product_code) as num
FROM ( SELECT DISTINCT section_code, section, para, subpara, chemical, product,
product_code, presentation FROM ebmdatalab.hscic.bnf )
GROUP BY chapter, section_code, presentation
HAVING num >1   --where name maps to more than one bnf code
ORDER BY chapter, num DESC)

SELECT
section_code, section, para, subpara, chemical, product, product_code,
presentation  -- this level is to filter to the top-prescribed code for each
drug name (according to latest detailed monthly data) (or, if none were
prescribed, then the first product name alphabetically
FROM (
  SELECT  -- this level joins all possible product codes to aggregated
prescribing data (2011-16) and ranks by items prescribed.
    a.*, b.items AS items_2011_2016,
    row_number() OVER (PARTITION BY a.presentation ORDER BY b.items DESC) AS
ranking -- We can use this to select the top/most likely drug code
  FROM (  -- this level is to look up all possible product codes for each drug
name in current BNF
      SELECT
```

```
      DISTINCT -- here we just want to go to product level rather than
individual presentations
      presentation,
      chapter, chapter_code,
      section, section_code,
      para, para_code,
      subpara, subpara_code,
      chemical, chemical_code,
      product, product_code
      from ebmdatalab.hscic.bnf where presentation in (select presentation from
temp where chapter < '18')

      ) a  -- now join to aggregated dataset grouped up to product level:
      LEFT JOIN ( SELECT substr(bnf_code,1,11) AS product_code, sum(items) as
items from ebmdatalab.aggregated_data.all_prescribed_BNFs_UpToSept2016 GROUP BY
product_code ) b
            ON a.product_code = substr(b.product_code,1,11)
      )
WHERE ranking = 1
ORDER BY chapter_code, presentation, product_code
```

A2. The lookup table of alternative drug spellings found within the data is created using the script below and saving as **ebmdatalab.hscic.drug_name_alt_spellings_in_PCA_data_HC**

```
-- find drug name changes in PCA data to 2016
-- save results as ebmdatalab.hscic.drug_name_alt_spellings_in_PCA_data_HC

WITH
a AS  (
  SELECT
      IF(LENGTH(bnf_7_char)=9,SUBSTR(bnf_7_char,2,4),SUBSTR(bnf_7_char,1,4)) AS
section_code, -- extra clause added to deal with those with extra spaces
2017-08-01

SUBSTR(drug_name,1,IF(STRPOS(drug_name,'_')>0,STRPOS(drug_name,'_')-1,length(dr
ug_name))) AS drug_name_part,  --take first part of drug name, up to underscore
(if there is one)
      MIN(year) AS min_year,  --this will help us to see which are the older vs
newer spellings used
      MAX(Year) AS max_year,
      SUM(items) AS Items
  FROM ebmdatalab.hscic.prescribing_pca_1998_2016_full
  WHERE  IF(LENGTH(bnf_7_char)=9,SUBSTR(bnf_7_char,2,2),SUBSTR(bnf_7_char,1,2))
< '18'
  GROUP BY
```

```
  section_code,
  drug_name_part
  ),

 b AS
(SELECT DISTINCT
section_code,
drug_name_part,
REPLACE(REPLACE(drug_name_part, 'i', '__'),'y','__') AS IY,
REPLACE(REPLACE(drug_name_part, 's', '__'),'z','__') AS SZ,
REPLACE(REPLACE(drug_name_part, 'ph', '__'),'f','__') AS PHF,
CONCAT(drug_name_part,'e') AS E,   -- add and E on to the end (note this only
works for the LAST word)
REPLACE(drug_name_part, ' ', ' e') AS E_mid -- add an E on to the end of all
words occurring before a space
FROM a)


SELECT
a.section_code,
a.drug_name_part,
CAST(a.min_year AS STRING) AS start_date,
CAST(a.max_year AS STRING) AS end_date,
a.items,
b.drug_name_part AS alternative,
CASE WHEN REPLACE(REPLACE(a.drug_name_part, 'i', '__'),'y','__') = b.IY THEN
'i_y'
      WHEN  REPLACE(REPLACE(a.drug_name_part, 's', '__'),'z','__') = b.SZ THEN
's_z'
      WHEN  REPLACE(REPLACE(a.drug_name_part, 'ph', '__'),'f','__') = b.PHF
THEN 'ph_f'
      WHEN  a.drug_name_part = b.E OR CONCAT(a.drug_name_part,'e') =
b.drug_name_part THEN 'e_end'
      WHEN  a.drug_name_part = b.E_mid OR REPLACE(a.drug_name_part, ' ', 'e ')
=b.drug_name_part THEN 'e_end'
      END AS type
FROM a
INNER JOIN b
  ON (REPLACE(REPLACE(a.drug_name_part, 'i', '__'),'y','__') = b.IY
      OR REPLACE(REPLACE(a.drug_name_part, 's', '__'),'z','__') = b.SZ
      OR REPLACE(REPLACE(a.drug_name_part, 'ph', '__'),'f','__') = b.PHF
      OR a.drug_name_part = b.E   --note this will only show this match once,
so we put in the other way around also
      OR a.drug_name_part = b.E_mid
      OR CONCAT(a.drug_name_part,'e') = b.drug_name_part
      OR REPLACE(a.drug_name_part, ' ', 'e ') =b.drug_name_part)
  AND a.drug_name_part != b.drug_name_part
  AND a.section_code = b.section_code
```

```
ORDER BY items desc
```

A3. The lookup table of Chemical name changes is created using the script below and saving as **pca_chemical_old_to_new_lookup**

```
-- PCA data - finding up to date chemical to combine with dataset
--save results as ebmdatalab.hscic.pca_chemical_old_to_new_lookup
WITH A as (
  SELECT
    IF(LENGTH(bnf_7_char)=9,SUBSTR(bnf_7_char,2,4),SUBSTR(bnf_7_char,1,4)) AS
Section,
    drug_name,
    count(distinct chemical) AS chems,
    max(year) AS Max_year_overall
    FROM
    ebmdatalab.hscic.prescribing_pca_1998_2016_full
    where
IF(LENGTH(bnf_7_char)=9,SUBSTR(bnf_7_char,2,2),SUBSTR(bnf_7_char,1,2)) <'18'
    GROUP BY
    Section,
    drug_name
    HAVING chems >1
    ),

B AS (
SELECT IF(LENGTH(bnf_7_char)=9,SUBSTR(bnf_7_char,2,4),SUBSTR(bnf_7_char,1,4))
AS section, drug_name, chemical,
    min(year) AS Min_year,
    max(year) AS Max_year
    FROM ebmdatalab.hscic.prescribing_pca_1998_2016_full
GROUP BY Section, drug_name, chemical
    ),

C AS (
SELECT DISTINCT
    A.Section,
    A.drug_name,
    B.chemical,
    B.min_year,
    B.max_year,
    IF(max_year = Max_year_overall,1,0) AS latest
    FROM A LEFT JOIN B ON A.drug_name = B.drug_name AND A.Section = B.Section
ORDER BY drug_name, chemical
)

SELECT old.section, old.drug_name, old.chemical AS old_chemical_name,
    nw.chemical AS new_chemical_name, nw.min_year AS Since
    FROM c old
```

```
     LEFT JOIN c nw ON old.drug_name = nw.drug_name AND old.chemical !=
nw.chemical and nw.latest = 1
    WHERE old.latest = 0
ORDER BY old.section, old.drug_name
```

A4. Known drug name changes

As reported online by patient.info
`ebmdatalab.hscic.drug_name_changes_2013`

A5. Fuzzy lookup for drugs not matching to BNF

List of drugs not matching BNF, identified through earlier iterations of the code.
These 1,084 drugs were matched to similar BNF names via fuzzy lookup in Excel and manually checked by a pharmacist.

List available at:
https://docs.google.com/spreadsheets/d/1UweKIZOLrKEzCtLULk5R5kJ4UyFIttvEouQ7RFGYrEA/edit#gid=594622641
and stored as `ebmdatalab.hscic.pca_bnf_name_to_code_fuzzy_lookup`

*B - Data Extraction And Normalisation*

B1. The latest chemical name for each drug is appended into the full dataset, to create prescribing_pca_1998_2016_full_v2

This does not take into account spelling changes but those will be handled later

```
-- save results as ebmdatalab.hscic.prescribing_pca_1998_2016_full_v2

SELECT a.*, COALESCE(c2.new_chemical_name, a.chemical) AS new_chemical_name
FROM
ebmdatalab.hscic.prescribing_pca_1998_2016_full a

LEFT JOIN ebmdatalab.hscic.pca_chemical_old_to_new_lookup c2
   ON a.drug_name = c2.drug_name
   AND a.chemical = c2.old_chemical_name
   AND IF(LENGTH(bnf_7_char)=9,SUBSTR(bnf_7_char,2,4),SUBSTR(bnf_7_char,1,4)) =
c2.section
```

C2. Run final data extraction parts 1 and 2 (scripts copied and updated from Issues #6 and #7)

## B2a. Part 1

### -- Final PCA data extraction part 1 (2016)

```
-- save results as ebmdatalab.tmp_eu.trends_from_pca

WITH
temp AS
(SELECT DISTINCT X.section_code, X.drug_name_part AS old_name, X.alternative
FROM
ebmdatalab.hscic.drug_name_alt_spellings_in_PCA_data_HC X
INNER JOIN ebmdatalab.hscic.drug_name_alt_spellings_in_PCA_data_2016_HC Y ON
X.alternative = Y.drug_name_part AND Y.end_date = '2016'
),

b AS (
  SELECT DISTINCT
   chapter_code, chapter, section_code, section, para, subpara, chemical,
product, product_code,
   REPLACE(presentation,'GlucOsamine','prop-GlucOsamine') AS
presentation,REPLACE(presentation,' ','')
     AS presentation_no_spaces
   FROM ebmdatalab.hscic.bnf
   WHERE presentation NOT IN (SELECT presentation from
ebmdatalab.hscic.bnf_name_to_product_special_cases_helen)
   AND chapter_code <'18'),

a0 AS (
SELECT *,
      TRIM(bnf_7_char) AS bnf_7_char_trim,

SUBSTR(drug_name,1,IF(STRPOS(drug_name,'_')>0,STRPOS(drug_name,'_')-1,length(dr
ug_name)))
        AS drug_name_part,
      SUBSTR(drug_name,1,IF(STRPOS(drug_name,' ')>0,STRPOS(drug_name,'
')-1,length(drug_name)))
        AS drug_name_part_short,
      SUBSTR(chemical,1,IF(STRPOS(chemical,' ')>0,STRPOS(chemical,'
')-1,length(chemical)))
        AS chemical_short,
      REPLACE(drug_name,'GlucOsamine','prop-GlucOsamine') AS drug_name_a,

REPLACE(REPLACE(drug_name,'GlucOsamine','prop-GlucOsamine'),'Sulph','Sulf') AS
drug_name_b,
      REPLACE(new_chemical_name,'Sulph','Sulf') AS new_chemical_name_b
```

```
      FROM ebmdatalab.hscic.prescribing_pca_1998_2016_full_v2 a
      WHERE
IF(LENGTH(bnf_7_char)=9,SUBSTR(bnf_7_char,2,2),SUBSTR(bnf_7_char,1,2)) < '18'),

a1 AS (SELECT a0.*,
        z.new_bnf_code AS code_fuzzy,
        z.new_name AS drug_name_fuzzy,
        CONCAT( UPPER(substr(d.new_name,1,1)),
substr(D.new_name,2,LENGTH(D.new_name)-1) ) AS product_2013,
        E.alternative AS product_new_spelling,
        CONCAT( UPPER(substr(d1.new_name,1,1)),
substr(D1.new_name,2,LENGTH(D1.new_name)-1) ) AS chemical_2013,  -- note, this
capitalises the first letter only
        replace(a0.new_chemical_name_b,a0.chemical_short,D3.new_name) AS
chemical_2013b,
        replace(a0.drug_name_b,a0.drug_name_part,e.alternative) AS
converted_drug_name, -- incorporate new spellings into drug name
        replace(a0.drug_name_b,a0.drug_name_part,D.new_name) AS
converted_drug_name2,
        replace(a0.drug_name_b,a0.drug_name_part_short,D2.new_name) AS
converted_drug_name3,

SUBSTR(drug_name_b,1,IF(STRPOS(drug_name_b,'_')>0,STRPOS(drug_name_b,'_')-1,len
gth(drug_name_b))) AS drug_name_part_b

        FROM a0
      LEFT JOIN ebmdatalab.hscic.drug_name_changes_2013 D ON
LOWER(drug_name_part) = D.old_name
      LEFT JOIN ebmdatalab.hscic.drug_name_changes_2013 D1 ON
LOWER(a0.chemical) = D1.old_name
      LEFT JOIN ebmdatalab.hscic.drug_name_changes_2013 D2 ON
LOWER(drug_name_part_short) = D2.old_name
      LEFT JOIN ebmdatalab.hscic.drug_name_changes_2013 D3 ON
LOWER(a0.chemical_short) = D3.old_name
      LEFT JOIN temp E ON a0.drug_name_part = E.old_name AND
SUBSTR(a0.bnf_7_char_trim,1,4) = e.section_code
      LEFT JOIN ebmdatalab.hscic.pca_bnf_name_to_code_fuzzy_lookup z ON
A0.drug_name = z.old_name
      ),


--CAPITALISE WHERE NEEDED:
A2 AS (
SELECT *,
      CONCAT( UPPER(substr(converted_drug_name2,1,1)),
substr(converted_drug_name2,2,LENGTH(converted_drug_name2)-1) ) AS
converted_drug_name4,
```

```
      CONCAT( UPPER(substr(converted_drug_name3,1,1)),
substr(converted_drug_name3,2,LENGTH(converted_drug_name3)-1) ) AS
converted_drug_name5,
      CONCAT( UPPER(substr(chemical_2013b,1,1)),
substr(chemical_2013b,2,LENGTH(chemical_2013b)-1) ) AS chemical_2013_c
--capitalise chemical as well.
      FROM A1
      ),

--COALESCE TO FORM "FINAL" NAMES
a3 AS (
 SELECT *,
COALESCE(converted_drug_name,converted_drug_name4,converted_drug_name5,drug_nam
e_fuzzy,drug_name_b)
   AS drug_name_F,
 COALESCE(product_new_spelling,product_2013,drug_name_part_b) AS
drug_name_part_F,
 COALESCE(chemical_2013, chemical_2013_c, new_chemical_name_b) AS chemical_F
 from A2
 ),

 --add a drug name field without spaces:
 a4 AS (
 select *, REPLACE(drug_name_F,' ','')  as drug_name_F_no_spaces
 from A3
 ),


a AS  (
  SELECT
      x.bnf_7_char_trim AS bnf_code,
      x.drug_name,
      drug_name_F,
      COALESCE(spc.presentation,b.presentation,ba.presentation,
bb.presentation,bc.presentation, bd.presentation)
        AS current_bnf_name,
      COALESCE(spc.product_code,b.product_code,ba.product_code,
bb.product_code,bc.product_code, bd.product_code)
        AS current_bnf_code,
      drug_name_part,
      drug_name_part_F, -- use as product name if no other
      x.section,
      x.subpara,
      x.chemical AS Chemical_original,
      x.chemical_F AS Chemical,
      x.Year,
      SUM(x.owc2) AS OWC2,   -- prescribed generically but no generic available
      SUM(x.NIC) AS Cost,
```

```
      SUM(x.items) AS Items,
      SUM(x.quantity) AS Quantity
  FROM a4 x
    --AND A.Currently_in_BNF = 'N'
    LEFT JOIN ebmdatalab.hscic.bnf_name_to_product_special_cases_helen spc ON
upper(x.drug_name_F) = upper(spc.presentation)  -- look up original drug
details in current bnf (drugs matching more than one drug in bnf)
    LEFT JOIN b ON upper(x.drug_name_F) = upper(b.presentation)   -- use upper
to match up examples like this: "Pentasa Sr_Tab 250mg" and "Pentasa SR_Tab
250mg"
            AND SUBSTR(x.bnf_7_char_trim,1,4) = b.section_code   -- look up
original drug details in current bnf.

    LEFT JOIN b ba ON upper(x.drug_name_F) = upper(ba.presentation)
            AND SUBSTR(x.bnf_7_char_trim,1,4) != ba.section_code  -- check if
drug now only belongs in a different section but same chapter
            AND SUBSTR(x.bnf_7_char_trim,1,2) = ba.chapter_code
    LEFT JOIN b bb ON upper(x.drug_name_F) = upper(bb.presentation)
            AND SUBSTR(x.bnf_7_char_trim,1,2) != bb.chapter_code  -- check if
drug now only belongs in a different chapter
    LEFT JOIN b bc ON upper(x.drug_name) = upper(bc.presentation) AND
b.presentation IS NULL AND ba.presentation IS NULL AND bb.presentation IS NULL
--also check original in case new drug name didn't work e.g. nifedipin(e)
            AND SUBSTR(x.bnf_7_char_trim,1,4) = bc.section_code
    LEFT JOIN b bd ON x.drug_name_F_no_spaces = bd.presentation_no_spaces   --
match without spaces e.g. Terbut Sulf_Inha 250mcg (400 D) vs "(400D)"
            AND SUBSTR(x.bnf_7_char_trim,1,4) = bd.section_code  AND
b.presentation IS NULL AND ba.presentation IS NULL AND bb.presentation IS NULL
-- look up original drug details in current bnf.

  GROUP BY
      bnf_code, drug_name, drug_name_F, current_bnf_name, current_bnf_code,
      drug_name_part, drug_name_part_F, -- use as product name if no other
      section, subpara, Chemical_original, Chemical, Year
      )

SELECT
    a.bnf_code,
    a.current_bnf_code AS Product_code_updated,
    SUBSTR(COALESCE(a.current_bnf_code,b.product_code,a.bnf_code),1,2) AS
Chapter_code_current,
    SUBSTR(a.bnf_code,1,2) AS BNF_Chap_Code,
    COALESCE(b.chapter, ch.description) AS Chapter_Current,
    ch.description AS Chapter_original,
    SUBSTR(COALESCE(a.current_bnf_code,a.bnf_code),3,2) AS
Section_code_current,
    SUBSTR(bnf_code,3,2) AS BNF_Section_Code,
    COALESCE(b.section,se.description,a.section) AS Section_Current,
```

```
    a.section AS Section_Original,
    SUBSTR(COALESCE(a.current_bnf_code,b.product_code,a.bnf_code),5,2) AS
Para_code_current,
    COALESCE(b.para,pa.description) As Para_current,
    COALESCE(b.subpara,a.subpara) As Subpara_current,
    a.subpara AS Subpara_original,
    COALESCE(b.chemical,a.chemical) As Chemical_current,
    a.Chemical_original,
    COALESCE(b.product, a.drug_name_part_F) AS Product_current,
    current_bnf_name,
    a.drug_name,
    IF(b.product_code IS NULL,'N','Y') AS Currently_in_BNF,
    a.year,
    a.Items,
    a.owc2,
    a.Quantity,
    a.Cost

  FROM a
  LEFT JOIN ebmdatalab.hscic.bnf_vertical ch ON SUBSTR(a.bnf_code,1,2) =
ch.code
  LEFT JOIN ebmdatalab.hscic.bnf_vertical se ON SUBSTR(a.bnf_code,1,4) =
se.code
  LEFT JOIN ebmdatalab.hscic.bnf_vertical pa ON SUBSTR(a.bnf_code,1,6) =
pa.code
  LEFT JOIN b ON a.current_bnf_name = b.presentation
            AND a.current_bnf_code = b.product_code
Save results as ebmdatalab.tmp_eu.trends_from_pca_2016
```

## B2b. Part 2

```
-- final pca data extraction (2016) part 2
-- distinct product-chemical combinations in current BNF:
WITH
 chem_p AS (
   SELECT DISTINCT product, product_code, chemical_code, chemical,
   count (distinct product_code) Over (partition by chemical_code, product)
      AS Dist_prods_with_same_name
   FROM ebmdatalab.hscic.bnf
   WHERE chapter_code <'18'
   ORDER BY Dist_prods_with_same_name, product),

-- find all drug_name_parts in PCA which have been mapped to a new chemical:
chem_0 AS (
  SELECT
```

```
SUBSTR(drug_name,1,IF(STRPOS(drug_name,'_')>0,STRPOS(drug_name,'_')-1,length(dr
ug_name)))
      AS drug_name_part,
  drug_name, section, old_chemical_name, new_chemical_name
  FROM ebmdatalab.hscic.pca_chemical_old_to_new_lookup_2016),

-- distinct *chemicals* in current BNF:
chem_a AS (
SELECT chemical,
    count(distinct chapter) AS Chapters,
    count(distinct section) AS Sections,
    count(distinct para) AS Paras,
    count(distinct chemical_code) AS Codes,
    min(chemical_code) AS min_code
  FROM ebmdatalab.hscic.bnf
  WHERE chapter_code < '18'
  GROUP BY chemical
  ORDER BY codes DESC, paras DESC, chemical),

-- for chemicals with multiple codes:
-- check whether each chemical code is the only one in its paragraph / section
/ chapter
chem_a1 AS
(SELECT DISTINCT
  a.chemical, b.chemical_code, a.paras, b.para_code, a.sections,b.section_code,
a.chapters, b.chapter_code,
  count(distinct b.chemical_code) over (partition by b.chemical,chapter_code)
    AS appearances_by_chapter,
  count(distinct b.chemical_code) over (partition by b.chemical,section_code)
    AS appearances_by_section,
  count(distinct b.chemical_code) over (partition by b.chemical,para_code)
    AS appearances_by_para
  FROM ebmdatalab.hscic.bnf b
  INNER JOIN chem_a a ON a.chemical = b.chemical and a.codes > 1
  WHERE b.chapter_code < '18'
  ORDER BY chemical ),


-- SELECT ALL CHEMICALS FROM BNF WHICH MAP TO A SINGLE PRODUCT
-- used in final step only
b AS (
  SELECT DISTINCT
   chapter_code, chapter, section_code, section, para_code, para, subpara_code,
subpara, chemical_code
   FROM ebmdatalab.hscic.bnf
   WHERE chapter_code <'18'),
```

```
t as (
 SELECT t.*,
 REPLACE(c2.new_chemical_name,'Streptokinase-Streptodornase','Streptokinase &
Streptodornase')
      AS new_chemical_name,
 CASE WHEN Product_current LIKE 'Levonelle%' THEN '0703050A0BC'   --
'Levonelle'
      WHEN Product_current LIKE 'Postinor%' THEN '0703050A0BB'    -- 'Postinor'
      WHEN t.drug_name LIKE 'Terbut%Sulph_Syr%' THEN '0301011V0AA'  -- 'Terbut
Sulf'
      WHEN t.drug_name LIKE 'Thalidomide%' AND Chapter_code_current = '05' THEN
'0501100J0AA' -- 'Thalidomide (Antileprotic)'
      WHEN Product_current LIKE 'Menoring 50' THEN '0702010G0BE'  -- 'Menoring
50'
      WHEN t.drug_name = 'Acetylcy_Eye Dps 10% (Old)' THEN '1108010C0AA' --
'Acetylcy (Eye)'
      WHEN t.drug_name = 'Abilify Maintena_Inj 400mg Vl + Dil' THEN
'0402020ADBB' -- 'Abilify Maintena'
      WHEN Product_current LIKE 'Melatonin%' THEN '0401010ADAA' -- 'Melatonin'
      WHEN Product_current LIKE 'Varidase%' THEN '1311070R0BB' -- 'Varidase'
      WHEN t.drug_name = 'Cocois_Scalp Oint' THEN '1305020V0BB' -- 'Cocois'
      WHEN t.drug_name = 'Levocarnitine_Oral Soln Paed 1.5g/5ml30%' THEN
'0908010C0AA' -- 'Levocarnitine'
      ELSE product_code_updated
      END AS product_code_updated_manual,
 CASE WHEN Product_current LIKE 'Levonelle%' THEN 'Levonelle'
      WHEN Product_current LIKE 'Postinor%' THEN 'Postinor'
      WHEN t.drug_name LIKE 'Terbut%Sulph_Syr%' THEN 'Terbut Sulf'
      WHEN t.drug_name LIKE 'Thalidomide%' AND Chapter_code_current = '05' THEN
'Thalidomide (Antileprotic)'
      WHEN Product_current LIKE 'Menoring 50' THEN 'Menoring 50'
      WHEN t.drug_name = 'Acetylcy_Eye Dps 10% (Old)' THEN 'Acetylcy (Eye)'
      WHEN t.drug_name = 'Abilify Maintena_Inj 400mg Vl + Dil' THEN 'Abilify
Maintena'
      WHEN Product_current LIKE 'Melatonin%' THEN 'Melatonin'
      WHEN Product_current LIKE 'Varidase%' THEN 'Varidase'
      WHEN t.drug_name = 'Cocois_Scalp Oint' THEN 'Cocois'
      WHEN t.drug_name = 'Levocarnitine_Oral Soln Paed 1.5g/5ml30%' THEN
'Levocarnitine'
      ELSE product_current
      END AS product_current_manual
 FROM ebmdatalab.helen.trends_from_pca_2016 t
 LEFT JOIN chem_0 c2 ON t.drug_name = c2.drug_name AND t.chemical_current =
c2.old_chemical_name AND SUBSTR(t.bnf_code,1,4) = c2.section
 ),
```

```
  A AS (
 SELECT T.*,
 COALESCE(chem_p.product,c2.product,product_current_manual) AS current_product,
    -- use this order in coalesce because we want to update/replace any
existing product names for which we now have a better one.
 COALESCE(product_code_updated_manual,chem_p.product_code,c2.product_code)
    AS current_product_code,

COALESCE(chem_p.chemical,c2.chemical,chem_a.chemical,chem_a1.chemical,c3.chemic
al)
    AS unique_chem,  -- chemicals currently in BNF (uniquely)

COALESCE(chem_p.chemical_code,c2.chemical_code,chem_a.min_code,chem_a1.chemical
_code,c3.min_code)
    AS unique_chem_code

 FROM  t
  -- link to BNF using whole Product name (note this will be drug_name_part)
--------------------
  -- chemical must match as well because product names are not always unique.
  LEFT JOIN chem_p ON t.product_current = chem_p.product
            AND t.Product_code_updated_manual IS NULL
            AND (UPPER(chem_p.chemical) = UPPER(Chemical_current)
                      OR  UPPER(chem_p.chemical) = UPPER(new_chemical_name))
            AND SUBSTR(chem_p.chemical_code,1,6) = SUBSTR(bnf_code,1,6)
                      -- some chemicals sit in multiple paras.
            AND chem_p.Dist_prods_with_same_name = 1

  -- try shortening Product names in BNF to match products in data (only if
whole name is not found) --
    LEFT JOIN chem_p c2 ON t.product_current =
SUBSTR(c2.product,1,length(t.product_current))
            AND t.Product_code_updated_manual IS NULL
            AND chem_p.product IS NULL
            AND UPPER(c2.chemical) IN (UPPER(Chemical_current),
UPPER(new_chemical_name))
            AND SUBSTR(c2.chemical_code,1,6) = SUBSTR(bnf_code,1,6)  --some
chems sit in multiple paras.
            AND chem_p.Dist_prods_with_same_name = 1

  -- link to BNF using "original" chemical name for chemicals which are unique
in BNF  ---------------
  LEFT JOIN chem_a ON UPPER(chem_a.chemical)  = UPPER(Chemical_current)
            AND chem_a.codes = 1 AND chem_p.chemical IS NULL
            AND t.Product_code_updated_manual IS NULL
  -- link to BNF using NEW chemical name for chemicals which are unique in BNF
--------------------
  LEFT JOIN chem_a c3 ON c3.chemical  = new_chemical_name
```

```
            AND c3.codes = 1 AND chem_a.chemical IS NULL
            AND chem_p.chemical IS NULL
            AND t.Product_code_updated_manual IS NULL
  -- link to BNF using NEW chemical name for chemicals which are NON-unique in
BNF  --------------
  -- provided that no chemical has been assigned in a previous join.
  -- first check same paragraph then section then chapter.
  LEFT JOIN chem_a1 ON chem_a1.chemical  = Chemical_current
            AND chem_p.chemical IS NULL
            AND chem_a.chemical IS NULL
            AND c3.chemical IS NULL
            AND t.Product_code_updated_manual IS NULL
            AND (
              (chem_a1.para_code = SUBSTR(bnf_code,1,6) AND
chem_a1.appearances_by_para = 1)
              OR (chem_a1.section_code = SUBSTR(bnf_code,1,4) AND
chem_a1.appearances_by_section = 1)
              OR (chem_a1.chapter_code = SUBSTR(bnf_code,1,2) AND
chem_a1.appearances_by_chapter = 1)
                     )
 ORDER BY drug_name,year ),

 u AS (
 select bnf_code,
 Chapter_code_current,   BNF_Chap_Code,     Chapter_Current, Chapter_original,
 Section_code_current,   BNF_Section_Code,  Section_Current,   Section_Original,
 Para_code_current,      Para_current,
 Subpara_current,  Subpara_original,
 COALESCE(unique_chem_code, SUBSTR(Product_code_updated,1,9),
SUBSTR(current_Product_code,1,9))
   AS chem_code_today, --chemical code
 Chemical_original,
 COALESCE(unique_chem,Chemical_current) AS chem_today,
 COALESCE(Product_code_updated, current_product_code)    AS prod_code_today,
 COALESCE(current_product,     Product_current) AS prod_today,
   -- note this is opposite way around to code because we want to replace the
previous name
   -- but there may not be a code.
 current_bnf_name, drug_name,
 Currently_in_BNF,
 year,  Items,  owc2,  Quantity,  Cost
 FROM a
 ORDER BY drug_name, year)


SELECT
    bnf_code,
    COALESCE(b.chapter_code,Chapter_code_current) AS Chapter_code_current,
```

```
    BNF_Chap_Code,
    COALESCE(b.chapter,Chapter_Current) AS Chapter_Current,
    Chapter_original,
    COALESCE(SUBSTR(b.section_code,3,2),Section_code_current) AS
Section_code_current,
    BNF_Section_Code,
    COALESCE(b.section,Section_Current) AS Section_current,
    Section_Original,
    COALESCE(SUBSTR(b.para_code,5,2),Para_code_current) AS Para_code_current,
    COALESCE(b.para,Para_current) AS Para_current,
    COALESCE(b.subpara,Subpara_current) AS Subpara_current,
    Subpara_original,
    chem_code_today AS Chemical_code_current,
    Chemical_original,
    chem_today AS Chemical_current,
    prod_code_today AS Prod_code_current,
    prod_today AS product_current,
    current_bnf_name, drug_name,
    Currently_in_BNF, u.year,  Items,  owc2,  Quantity,  Cost,
    -- add calculated fields:
    1000*items/pop.Population AS ItemsPer1000,
    1000*quantity/pop.Population AS QuantityPer1000,
    Inf.Multiplier_2016*cost AS Infl_corr_Cost,
    1000*Inf.Multiplier_2016*cost/pop.Population AS Infl_corr_Cost_per1000,
    IEEE_DIVIDE(Inf.Multiplier_2016*Cost, Items) AS Infl_corr_CostPerItem,
    1000*owc2/pop.Population AS Owc2Per1000

 FROM U
 LEFT JOIN b ON u.chem_code_today = b.chemical_code
 LEFT JOIN ebmdatalab.ONS.england_midyear_population pop ON u.Year = pop.Year
 LEFT JOIN ebmdatalab.ONS.inflation_cpi inf ON u.Year = inf.Year
 --WHERE LENGTH(chem_code_today) =8
```